Deriving Semantic Sensor Metadata from Raw Measurements

Jean-Paul Calbimonte[1], Zhixian Yan[2], Hoyoung Jeung[3], Oscar Corcho[1], and Karl Aberer[2]

[1]OEG, Facultad de Informática,Universidad Politécnica de Madrid, Spain
jp.calbimonte@upm.es,ocorcho@fi.upm.es
[2] LSIR, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
zhixian.yan@epfl.ch,karl.aberer@epfl.ch
[3] SAP Research, Brisbane, Australia
hoyoung.jeung@sap.com

Abstract. Sensor network deployments have become a primary source of big data about the real world that surrounds us, measuring a wide range of physical properties in real time. With such large amounts of heterogeneous data, a key challenge is to describe and annotate sensor data with high-level metadata, using and extending models, for instance with ontologies. However, to automate this task there is a need for enriching the sensor metadata using the actual observed measurements and extracting useful meta-information from them.
This paper proposes a novel approach of characterization and extraction of semantic metadata through the analysis of sensor data raw observations. This approach consists in using approximations to represent the raw sensor measurements, based on distributions of the observation slopes, building a classification scheme to automatically infer sensor metadata like the type of observed property, integrating the semantic analysis results with existing sensor networks metadata.

1 Introduction

Ubiquitous sensor networks are a primary source of observations from the physical world, from environmental measuring stations, participatory or citizen sensing, to various sensor applications in traffic, media and health monitoring. Publishing sensor networks data on the web has the potential of increasing public awareness and involvement on these different domains at a massive scale [1]. Cheap sensing devices can be easily configured and deployed, plugged to sensor data platforms such as Cosm[1] for exploitation, storage and querying.

The increasing availability of sensor data in the web introduces higher heterogeneity, which makes it more difficult for potential users to make sense out of these data sources and be able to identify which ones are useful for their applications. An example of this scenario is the Swiss Experiment[2] project, a

[1] Cosm, formerly Pachube https://cosm.com/
[2] Swiss Experiment: http://www.swiss-experiment.ch/

platform that enables real-time publishing environmental data on the web, from a large-scale federation of sensor networks, mainly in the Swiss Alps. The published data is heterogeneous as it comes from different geographical locations, with different time spans (e.g. observations collected during 1 year, 3 months, etc.), as well as varying sampling rates (e.g. per minute, per 10 minutes). Moreover, the metadata for these sensor types is not always complete and coherent. As an example, to indicate that a sensor measures temperature (i.e. the *observed property*), different sensors use various tag names, like "temperature", "temp", "t", "msptemperature", "tp", etc. Although the data is available for anyone to use, these noisy descriptions are not understandable enough and do not provide semantic information about what this data is about.

In less-controlled scenarios than the Swiss Experiment, the problems of heterogeneity are even more noticeable. For instance in the Cosm web platform, users tag their sensor data as means of metadata, identifying which types of measurements they are publishing. Projects like the Air Quality Egg[3], aiming at promoting air-quality participatory sensing, enable almost any citizen to publish measurements at web-scale. However, the user-provided metadata is often incomplete. In many cases these tags are misleading or they are not provided at all, making it very hard for other users to query or make use of this data.

To overcome this problem, establishing explicit semantics on the metadata has been proposed in previous works, using sensor ontologies [2]. When using these ontologies, sometimes it is needed to manually map the semantic information from the sources to the new metadata model [3], which is a cumbersome and error-prone task. In this paper we propose a novel approach of semantic sensor analysis that infers semantic properties such as the type of observed property, using the raw sensor observations as input. The main contributions of this paper are the following:

- We propose a novel method for representing time series as distributions that represent the slopes of a linear approximation of the initial numeric sensor measurements.
- Based on the statistics of the observation slopes, we infer the type of observed property of the sensor measurements. We use a classification method that exploits the similarity of the slopes distributions.
- We provide a mechanism for enriching the sensor metadata, based on the SSN Ontology [2], with the metadata inferred from the observation slopes.
- We build a self-contained evaluation system linking raw sensor measurements to high-level semantics, and validate our method using two real-life environmental sensor datasets, from the Swiss Experiment and AEMET[4](the Spanish meteorological office).

The remainder of this paper is organized as follows: Section 2 describes the global approach proposed for semantic analysis of sensor data. Section 3 studies

[3] AirQuality Egg `http://airqualityegg.wikispaces.com/`
[4] Agencia Estatal de Metereología: `http://www.aemet.es`

the sensor data representation using slopes, whereas Section 4 focuses on building classification algorithm for inferring observed property types and integrating them to the sensor metadata. In Section 5, we experimentally evaluate our approach. Section 6 summarizes existing related work. Finally, Section 7 includes concluding remarks and points to future works.

2 From Raw Measurements to Semantic Metadata

Sensor data is typically represented as time series, describing the evolution over time of a certain observed property. Raw sensor data without any metadata that describes it, has limited use as it is hard to discover, integrate or interpret. While in controlled environments the sensor metadata can be reasonably well managed and controlled by the data owners, in the context of the sensor web, where any citizen is able to produce and publish data, it becomes a more difficult task. While semantic metadata has been shown to be effective for managing large sensor metadata repositories, current proposals require expensive manual curation and tagging (see Section 6). However, these approaches do not look into the data values, from which we can derive some of these metadata properties using analysis and mining techniques.

We describe in Figure 1 our architecture for deriving semantic metadata from sensor data measurements. The approach includes characterizing sensor time series and extracting their observed property types to enrich sensor metadata, and consists of four main layers:

- At the *sensor deployment* layer, sensor nodes provide initial measurements in terms of real-time numerical values, e.g. temperature, humidity, etc.
- In the *semantic sensor analysis* layer, we first represent the sensor data stream using linear approximations and calculate the observation slopes. Based on the sensor slopes, we are able to compute similarity between sensor data series, detecting the observed property types through classification, and performing detection of these types with partial information.
- A semantic representation of the analysis component is integrated into the *semantic metadata*. Using the SSN Ontology as a basis, and combined with domain specific ontologies, this enriched metadata is made available for further processing or querying.
- In the application layer, users can build tools and visualizations to query such sensor data and receive results that include the new metadata computed by the analysis layer.

The deployment layer is usually built using sensor or stream data management systems. These systems centralize the data captured by the devices and provide storage, query interfaces and streaming operators. As for the semantic metadata, we built upon previous work on semantic management of sensor networks [3], centered on the use of the SSN Ontology, coupled with domain ontologies and vocabularies for quantities and units of measurements. For the

analysis of the time series, we propose a representation based on the slopes of a linear approximation of the data, as described in Section 3. Then these representations can be used to compare and find similarities among new and existing time series, classifying them according to the detected observed property type, etc. As a result, we are able to complete and query the sensor metadata, as detailed in Section 4.

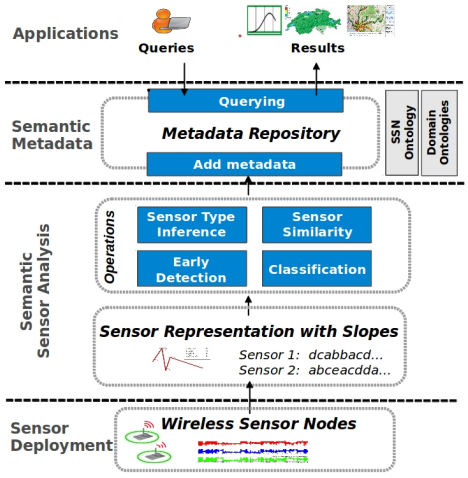
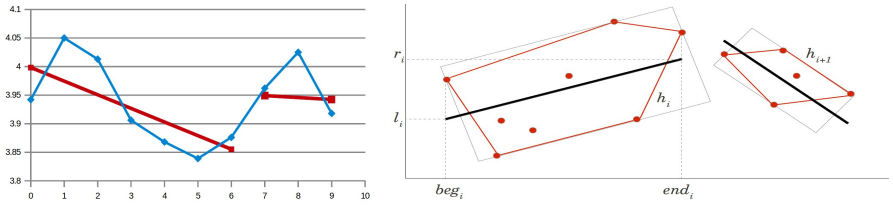


Fig. 1: Semantic Sensor Analysis Architecture

3 Sensor Data Representation with Slopes

In environmental time series, similar patterns can be observed periodically over time. These patterns can be characteristic to a type of sensor data, and therefore help to recognize it. If we represent a time series using a linear representation, such as the one in Figure 2(b), the patterns of the data can be associated to the angles of the linear segments or its corresponding slope. For instance, a steep slope indicates a sudden increase of the measured property. The intuition is that if these slopes are repetitive over time, we can build slope distributions that can be representative of a type of time series. Using slopes makes it possible to find similarities between time series that not necessarily have the same value ranges but similar behavior, e.g such as the *air temperature* in two different locations.



(a) Linear approximation (b) Constructing the convex hulls and segments

Fig. 2: Piecewise linear representations

3.1 Piecewise Linear Representation

We can use linear segments to approximate a time series (Piecewise Linear Representation, PLR), and analyze the trends by observing the angles that the segments form. For instance in Figure 2(a), we use 2 segments to represent the original 10 data points. Notice that the number of points for a segment can be variable (adaptive approximations). We used the algorithm of [4] for the construction of piecewise linear histograms.

Consider we have a time series of n data points $X = x_1, x_2, ..., x_n$, and we want to fit it in $m << n$ segments. The algorithm maintains a set B of buckets $b_i = h_i, beg_i, end_i, l_i, r_i, h_i$, where h_i is a convex hull of data points, and $(beg_i, l_i), (end_i, r_i)$ are the coordinates of the segment that best fits the convex hull (the segment that bisects the thinnest bounding rectangle of h_i [4]). The slope of b_i can be calculated as $slope(b_i) = \frac{r_i - l_i}{end_i - beg_i}$. The algorithm adds elements to B from X, until there are no buckets available, and then it starts to merge those adjacent buckets b_i and b_{i+1} that combined produce the smallest increase in total error. Merging is reduced to a convex hull merge of h_i and h_{i+1}. The algorithm iterates until all elements of X have been placed in a bucket. The resulting set of segments of each bucket b_i is the linear approximation of X.

For instance in Figure 2(b), the convex hull h_i encloses 8 data points and its minimum rectangle is bisected by the thick black segment defined by the points $(beg_i, l_i), (end_i, r_i)$. This is the linear representation for these 8 points. During the computation of the linear representation, if merging h_i with the next hull h_{i+1} reduces the approximation error, they will form a new single hull with its own bisecting segment. Once we apply this PLR algorithm we have the time series represented as line segments, each with a distinctive slope.

3.2 Slope Distributions

To build the slope distributions, we first compute a linear approximation of the time series, using the algorithm described in Section 3.1. It is possible to create linear approximations of different accuracy, depending on the number of segments per unit of time. For instance for a time series of 30 days, if we use 4 segments per day, their slopes will reflect coarse-grained changes in the data during each day. Time series of originally different sampling times, can be represented using the same segment/day rate, in order to be comparable. Obviously, if the original sampling interval is greater than the number of segments/day, the representation with that rate is not possible.

Once the linear representation is built, we can compute the slopes and analyze them. The slope or gradient space, bounded in the $[\infty, -\infty]$ interval for the possible angles $[\frac{\pi}{2}, -\frac{\pi}{2}]$, can be divided in sectors, each represented with a symbol α_j from an alphabet A and we can assign each segment to its corresponding symbol. We propose using the segment representation discussed in the previous section, to compute *slope symbolizations*, which characterize a time series as a sequence S of symbols s_i from an alphabet A that correspond to a type of slope.

In this way, we characterize a time series by the type of variations present in the sensor data, regardless of the data values. For example if we divide the angle space in 4 sectors (labeled a, b, c, d), at intervals of $\frac{\pi}{4}$, we can match each segment slope with one symbol. For instance in Figure 3 we have 4 segments, whose symbolic representation is *adac*, by matching each slope with a symbol.

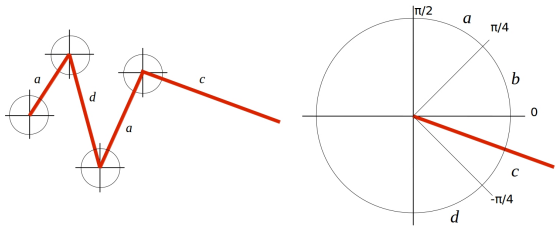


Fig. 3: Slopes symbolization. The angle space in this example is divided in 4 sectors, each of $\frac{\pi}{4}$. According to which division the segments falls in, it is assigned a symbol.

Having this symbolic representation of the slopes, it is possible to compare them to check if two series have similar slope patterns. One simple way to do so, is to generate *symbol distributions*, or histograms that count how many symbols of each type exist in a time series. So a distribution of a sequence S can be defined as a set D_S of elements $d_{\alpha_j} = |\{s_i \in S, s_i = alpha_j\}|$, for all symbols in A. For the previous example, it would be a vector $2, 0, 1, 1$, which can be normalized by the total elapsed time, so that we can compare series encompassing different time spans. A simple distance measure is the euclidean distance, defined for two distributions D_{S_1}, D_{S_2} of length n as: $d_{eucl}(D_{S_1}, D_{S_2}) = \sqrt{\sum_i^n (d_{S_1 i} - d_{S_2 i})^2}$

3.3 Choosing the angle divisions

Although we can arbitrarily choose how to divide the angle space (e.g. 4 sectors of $\frac{\pi}{4}$ as in the previous example), the actual angles may be more concentrated in some intervals than others. For instance time series with highly changing angles such as *wind speed*, may have steeper gradients than a more stable series. Taking into account this fact, we propose to analyze the training data sets to determine an angle division that better represents the actual distribution of angles in the training set. Using this distribution information, we can divide the angle space in divisions that hold the same number of angles of the training data.

4 Deriving Semantic Metadata

After establishing how the data is segmented and symbolized, we can use the symbol distributions for data analysis tasks to help understanding the semantics of the data. Given a time series, if it does not contain appropriate metadata, the potential user of this data can use already analyzed time series and compare the new one with them. We show how this can be done using our symbolization and a simple classification scheme, even with a partial subset of a time series.

4.1 Semantic Descriptions

A semantic description of an observation is a collection of statements that includes the observed property (e.g. humidity, pressure), feature of interest (e.g. the air at some location), unit of measurement, among others. For instance, using the vocabulary of the SSN Ontology [2], we describe a *wind speed* observation in Listing 1. The observation, identified as `swissex:WindSpeedObservation1`, has been observed by sensor `swissex:SensorWind1` and reported a value of `6.245`. The sensor observed property type `cf-property: wind_speed` (speed of the wind feature) is defined in a domain specific vocabulary (in this case the Climate and Forecast vocabulary defined by the W3C SSN-XG group[5]). Additional metadata about this observation are omitted for brevity.

```
swissex:WindSpeedObservation1 rdf:type ssn:Observation;
  ssn:featureOfInterest cf-feature:wind;
  ssn:observedProperty  cf-property:wind_speed;
  ssn:observationResult
   [rdf:type ssn:SensorOutput;
    ssn:hasValue [qudt:numericValue "6.245"^^xsd:double]];
    ssn:observedBy swissex:SensorWind1;
```

Listing 1: Wind Speed observation in RDF according to the SSN ontology

Concretely, the `cf-property:wind_speed` property indicates that this is an observation of wind speed, and it has further semantic information in the Climate & Forecast ontology, as seen in Listing 2. It states that it is a property of the wind (`cf-feature:wind`) and is a property of the more general *speed* quantity (`qu:speed`). In order to extract this information, the type of observed property from an unannotated dataset, we propose the classification scheme in the next subsection. The goal is basically to identify the `ssn:observedProperty` for a time series.

```
cf-property:wind_speed rdf:type dim:VelocityOrSpeed;
  rdfs:label "wind speed";
  ssn:isPropertyOf cf-feature:wind;
  qu:propertyType qu:scalar;
  qu:generalQuantityKind qu:speed.
```

Listing 2: Wind Speed property according to the Climate and Forecast vocabulary

4.2 Data Classification

Given two sets of time series, a *training set* already annotated according to the type of data that is captured, and an unannotated *test set*, we are interested in finding the observed property for the second set. Assume we have a collection $\mathcal{D}$ of symbol distributions $D_1, ..., D_i, ..., D_n$ as a training set, each of them corresponding to a time series ts_i, already classified with a type observed property (e.g. "wind speed"). The classification task consists in finding the best property for time series ts_{test} in the test set.

We can use a simple k-nearest neighbor scheme, which has been successfully used for time series classification [5,6]. First, the time series ts_{test} is segmented

[5] C&F vocabulary: `http://purl.oclc.org/NET/ssnx/cf/cf-property`

and symbolized. Then, we generate a symbol distribution D_{test}, as described in Section 3.2, which can be compared iteratively with each of the distributions D_i in $\mathcal{D}$. From the k distributions closer to D_{test}, we select the observed property of the majority.

4.3 Using Partial Data Subsets

This classification technique may use all the complete time series for computing the symbolization and the slope distribution. However, for types of data with recurring patterns such as the ones present in environmental and meteorological data, using a smaller subset of data can be enough to extract the feature that help detecting the type of observed property. In that case for the construction of the linear representation of the data, we simply choose a subset of the original data: $X = x_1, x_2, ..., x_n$, with a different n' such that $n' < n$.

4.4 Querying using the Analysis Results

After executing the classification, we can use the extracted information to complete the sensor metadata, that is then available for querying. In Listing 4 we show a simple SPARQL query that asks for sensors that measure air temperature.

```
SELECT ?sensor
WHERE {
 ?sensor a ssn:Sensor;
         ssn:observes cf-property:air_temperature.}
```

Listing 3: Query all sensors that measure air temperature

The streams produced by sensors can be seen as streaming datasets, whose metadata can also be queried. The stream, identified by a URI, can be seen as an unbounded dataset of observations, some of which are actually used to compute the slope symbolizations and classification described above. The observed properties obtained for the sensor (e.g. `cf-property:air_temperature`) are therefore the observed properties of the stream observations. We can also query for more general types of data, for instance, the generic *temperature* property. In Listing 4 we ask for all stream URIs of sensors that measure some type of temperature.

```
SELECT ?stream ?observedProperty
WHERE {
 ?sensor a ssn:Sensor;
         ssn:observes ?observedProperty.
 ?stream ssn:isProducedBy ?sensor.
 ?observedProperty qu:generalQuantityKind qu:temperature.}
```

Listing 4: Query all streams of sensors that measure air temperature

Furthermore, we can expose the similarity measurements computed between the time series, so that users can also query this information. As an example, in Listing 5 we use the Similarity Ontology[6](`sim`) to represent the computed distance between two series, using our slope representation. Then we can query, for instance the top 5 series similar to a given time series.

[6] The Similarity Ontology: `http://purl.org/ontology/similarity/`

```
swissex:slopeSim1_2 a sim:Similarity;
  sim:subject swissex:timeseries1;
  sim:object  swissex:timeseries2;
  sim:weight  0.32;
  sim:method  swissex:SlopeDistributionDistance.
```

Listing 5: Slope distribution similarity between two time series

This type of queries allows users not only to use the final results of a classification task, but also to query more detailed information including the precision of the computations. This information can be used to validate this metadata or provide insight about the analysis process and the relationship of a sensor stream with other streams. In the case of the early detection of the observed property of a time series, the user may be interested in knowing, for example, how many days of data are typically used for classifying those sensors that measure wind speed 6.

```
SELECT ?sensor ?dur
WHERE {
  ?sensor a ssn:Sensor;
          ssn:observes cf-property:wind_speed.
  ?timeseries ssn:isProducedBy ?sensor.
  ?timeseries swissex:duration [qu:numericalValue ?dur].}
```

Listing 6: Query the number of data days used for classifying wind speed sensors

5 Experimentation

The main goal of these experiments is to show that the proposed sensor data representation using slopes can be used to characterize sensor data and extract sensor metadata corresponding to the types of observed properties. First we show how the classification behaves with two real life data sets, in terms of precision. Next, we are interested in experimenting with smaller subsets of data samples, and observing how the classification behaves with less data, as we know there are repeating data patterns. Finally, we compare our approach with a classification using the widely used SAX symbolic representation of the data [5].

To validate the classification approach presented in Section 4.2, we implemented and applied it to two different datasets in the environmental domain: one from the Swiss Experiment[7] and another form AEMET. The data is heterogeneous as it comes from different geographical locations, some have different time spans (e.g. observations collected during 1 year, 3 months, etc), others have different sampling rates.Also the number of sensors per observation type varies (e.g. 78 for temperature, only 4 for snow height). Due to the conditions of the deployments, some of them experimental and others deployed in harsh environments, this dataset contains a considerable amount of noise in the data.

The AEMET dataset consists of sensor data from 100 weather stations managed by the Spanish meteorological office. The data is heterogeneous, coming from stations all over Spain, and was originally collected in intervals of 10 minutes. It contains, in general, less noise and anomalies than the Swiss Experiment dataset, as it comes from stations daily used for meteorological forecasts.

[7] The dataset is available at: `http://lsirpeople.epfl.ch/qvhnguye/benchmark/`

5.1 Classification in Swiss Experiment and AEMET

The goal of our first experiment consists in evaluating the effectiveness of the classification in terms of precision and recall. The classifier is expected to assign the correct label (the type of observed property, e.g. "humidity") to time series from a test set. The classifier uses a training set of time series and the evaluation criteria is computed in terms on the number of true positives (tp), false positives (fp) and false negatives (fn): precision ($p = \frac{tp}{tp+fp}$), and recall ($r = \frac{tp}{tp+tn}$).

Swiss Experiment The heterogeneity of the Swiss Experiment dataset required applying different parameters for the linear approximation step. Some time series had very short sampling time intervals (e.g. every 2 seconds for pressure, for at most two days), while others had very long ones (e.g. every half-an-hour for several months). Hence, the approximations were very different in these cases (hundreds of segments per day for short intervals, and only a few per day for long ones).We applied a 5-fold cross validation scheme to divide our dataset in training and test set, and then apply the nearest neighbor algorithm. We present the confusion matrix in Table 4, for $k = 5$.

Match results Swissex k=5, 5-fold

test set	ra	mo	te	wd	ws	hu	ly	pr	co	sh	vo	total	fp	tp	fn	p	r
radiation	15		4	7	7						1	34	19	15	0	0.441	1
moisture		10	3	2	1					1	1	20	8	10	2	0.556	0.833
temperature	2	3	56		1	11					2	78	19	56	3	0.747	0.949
wind direction	4		1	25	4							35	9	25	1	0.735	0.962
wind speed			1	4	40	1						46	6	40	0	0.87	1
humidity	1		9		2	21						34	12	21	1	0.636	0.955
lysimeter		2					4					6	2	4	0	0.667	1
pressure								4				4	0	4	0	1	1
co2									10			11	0	10	1	1	0.909
snow height		1	2							1		4	3	1	0	0.25	1
voltage			6		1						9	16	7	9	0	0.563	1
total												**288**	**85**	**195**	**8**	**0.7**	**0.96**

Fig. 4: Swiss Experiment confusion matrix, k=5. Column header abbreviations: ra:radiation, mo:moisture, te:temperature, wd:wind direction, ws:wind speed, hu:humidity, ly:lysimeter, pr:pressure, co:CO_2, sh: snow height, vo:voltage

We can observe that the effectiveness of the classification varies among the different types of data. The nearest neighbor scheme is also biased as the dataset is highly unbalanced. Since we have comparatively much more samples of temperature or wind speed, than for pressure or snow height, these last are less likely find nearest neighbors of the same class. For instance for *lysimeter* and *snow height*, almost no series are correctly identified, as we have a very small number of series. Nevertheless, in the cases of *pressure* or CO_2 the precision is good regardless of the low number of series. This is a special case, since these series have very different slope distributions, and also, have very short sampling interval. Since their resolution is much smaller (e.g. every 2 seconds) than most of the other series in the dataset, their comparison throws very large distances that are quickly discarded.

In cases where the total number of time series was very small (e.g. only 4 for *snow height*), the approach is clearly not effective. It requires a larger training set to have an acceptable precision. Also, when the series are very irregular (sometimes due to noise and false non-curated data in the original dataset), they logically fail to be correctly classified.

AEMET For the AEMET dataset, we followed the same approach as with the Swiss-Experiment. However, for the AEMET data, we had a larger number of time series for every type of data, thus avoiding the problem of lack of training data encountered in the previous tests. Moreover, the dataset sampling interval is the same, making it easier to compare their slope distributions. We applied the classification scheme with a 10-fold cross validation for this dataset. We provide the confusion matrix for $k = 5$ in Table 5.

Match results AEMET k=5, 10-fold

test set	st	ba	te	wd	ws	hu	wsx	pr	wdx	pre	total	fp	tp	fn	p	r
soiltemp	48		23	1	1	2		6			81	33	48	0	0.59	1
battery	2	66		2		1				10	81	15	66	0	0.81	1
temperature	15	1	84								100	16	84	0	0.84	1
winddirection		1		43	3				53		100	57	43	0	0.43	1
windspeed		1	1	2	54	4	37		1		100	46	54	0	0.54	1
humidity	1		1		2	92	2		2		100	8	92	0	0.92	1
windspeedmax		1	1	1	54	3	39		1		100	61	39	0	0.39	1
pressure			2					97			99	2	97	0	0.98	1
winddirmax		1		43	3				53		100	47	53	0	0.53	1
precipitation		2					1			97	100	3	97	0	0.97	1
total											**961**	**288**	**673**	**0**	**0.7**	**1**

Fig. 5: AEMET confusion matrix, k=5. Column header abbreviations: st:soil temperature, ba:battery, te:air temperature, wd:wind direction, ws:wind speed, hu:humidity, wsx: wind speed (max), pr:pressure, wdx: wind direction (max), pre:precipitation.

We can notice that in this case the approach achieves better precision, as expected, since we avoided the problems of sampling times and unbalanced types (the number of series per each type is similar or the same). However, it can be observed that there are important false positives at some specific spots. For instance the number of *soil temperature* series falsely identified as *air temperature* is very high. This is in fact an expected result, since both are specializations of the more general type *temperature*. Hence, both share patterns in the time series, that are reflected in the slope distributions that are compared during the classification process. The same situation can be seen between *wind speed* and *wind speed (max)*, and for *wind direction* and *wind direction (max)*.

It is also interesting to see that if we consider the "unification" of similar types of data (e.g. *wind speed* and *maximum wind speed*), the precision is much higher (Figure 6). This suggests that the slope distributions are useful for identifying similar data, because they have very similar slope distributions. This is an expected behavior, for instance for *wind speed* and *wind speed (max)*, which are measurements of the same type of data. In order to discern between

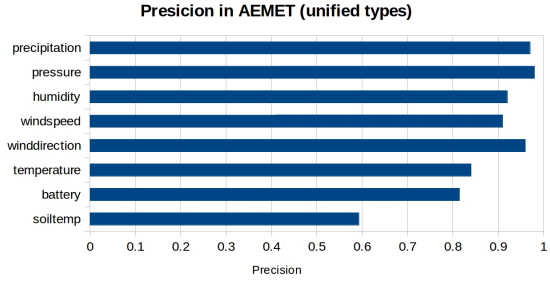
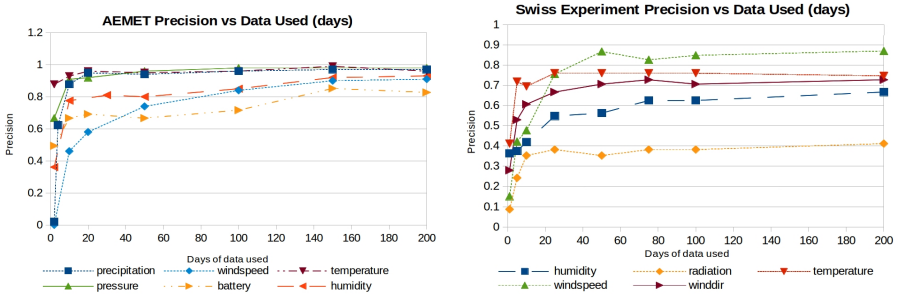


Fig. 6: Precision in AEMET, not differencing the specific types wind speed (max) and wind direction (max).

small differences like these, other characteristics of the data have to be taken into account. In these cases where two types of observations are similar, we can use a higher level definition of observed property. For instance, in the Climate and Forecast vocabulary, the specific properties `cf-property:air_temperature` and `cf-property:soil_temperature` both have `qu:temperature` as its general quantity kind.

5.2 Classification with Partial Information

In this experiment we aim at showing how the classification precision varies when using smaller subsets of the test data. As we discussed in Section 4.3, for our environmental and meteorological datasets, recurrent slope patterns in the data can be representative enough to compute the slope distribution, and make it possible to classify the data. We have tested the classification reducing the number of days-of-data used for computation. In Figure 7(a) and Figure 7(b) we plot the precision for the AEMET and Swiss Experiment dataset series, for different subsets of the data (expressed in terms of the number of days of measured data). In total we have around 200 days of observations, but we can see that for some types of data we require much less and obtain similar precision in the classification. This is the case especially with series that include very repetitive patterns on a daily basis, but not for others that have a more unpredictable behavior such as *wind speed*. In this case we see that it needs more days-of-data than other types to increase the precision.



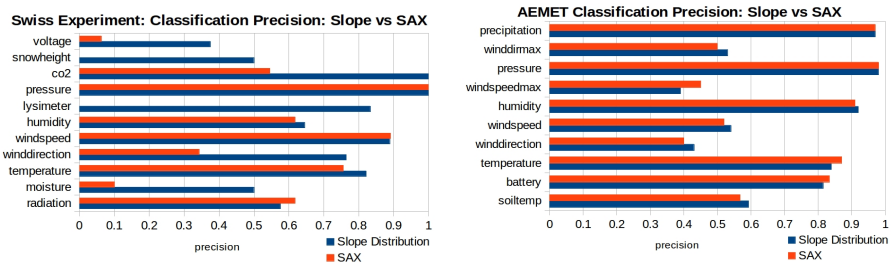
(a) AEMET (b) Swiss Experiment

Fig. 7: Classification precision, for different partial datasets, in terms of the days of data used.

5.3 Comparison with SAX Classification

The goal of this experiment is to compare our approach with a classification based on the widely used SAX representation of time series [5]. The comparison is based on the precision using both approaches. By classifying with SAX we can verify how well our method behaves in comparison to a well established technique. The SAX approach also produces a symbolization of the time series, although the angles and slopes are not taken into account, as it uses a PAA approximation. We applied the same classification method used for our slope-based

representation. We show the classification precision for the Swiss Experiment and AEMET datasets in Figure 8(a) and Figure 8(b) respectively.



(a) Swiss Experiment (b) AEMET

Fig. 8: Classification precision with SAX and the Slope representation.

As it can be seen, the classification throws similar results for both methods, with small differences in AEMET, and slightly better for the slope-based approach in the Swiss experiment dataset. Using the slopes distributions shows to be helpful at differencing time series with similar values but very different angles. In the case of AEMET, the measured values are already enough to discern between two different types of observation, and hence the results are not improved by the slope distribution. While the SAX representation has been exploited in other ways, for example by considering substrings of a fixed size, instead of only one symbol, this experiment shows that our approach is also able to extract features that help characterizing a type of time series, and enabling its semantic identification. A classification technique throws different results depending on the type of data. Further amendments could be plugged to the classification scheme, but they risk to be too specific to the characteristics of certain datatypes, and such methods are outside of the scope of this work.

6 Related Work

Previous works on time series classification and mining, have studied different approaches for summarizing and exploiting sensor raw data, and have been complemented with semantic representations for sensor data management.

Data Approximations High level representations reduce the dimensionality of time series data, in order to reduce the complexity of indexing and comparison algorithms, using different techniques. These include piecewise constant and linear approximations (e.g. PAA[7], APCA[8], PLR[9]) that use constant and linear segments respectively, to represent the original time series. Generally simple to compute, either in batch mode and online using sliding window algorithms, these methods offer accurate approximations of the original data. These representations have been widely used for tasks including similarity search, fuzzy queries, dynamic time warping, clustering and classification [9].

While these approximations reduce dimensionality, some approaches introduce a further step that consists in the symbolization of the time series. These

techniques, such as SAX [5], have shown to be space and time efficient for indexing, classification and clustering, and also for additional tasks such as motif discovery and visualization [10]. These symbolizations can be used to compute distance measures that help in classification and clustering tasks [5]. Other works have considered also the slopes of linear approximations such as the STS distance [11] for clustering time series.

SAX symbolization has also been used for sensor events detection [12] and for creating high-level perception abstractions from the raw sensor data, by matching SAX patterns with low-level thematic abstractions [13].

Time Series Classification Particularly, for the task of classification, different techniques such as decision trees, neural networks and bayesian classifiers have been used [6]. Classification approaches usually fall into the following three categories: distance-based, feature-based and model-based[6]. Simple distance measures such as euclidean, are very limited because they only consider one-to-one matches in the time axis. Distance measures with more elastic matching for the time axis, such as Dynamic Time Warping (DTW), have been proved effective for similarity matching [14]. These have been coupled with k-nearest neighbor (k-NN) classifiers, proving an effective combination for a number of time series classification problems [15,16]. These techniques have space and time computation limitations in some scenarios, and offer little explanation on why a series belongs to a particular class [17]. Feature-based approaches try to find properties that are representative of a type of series, in order to classify them. Most of these approaches use a high level representation e.g. symbolization or discretization methods, before extracting the features[6] while others work extracting representative subsequences (e.g. shapelets [17]).

Semantic Sensor Representations The task of modeling sensor data and metadata with ontologies has been addressed by the semantic web research community in recent years. Early ontology proposals for describing wireless sensors have been reviewed in [18]. However, the focus of most of these approaches was on sensor meta information, while the description of observations was generally overlooked. Besides some of these approaches lack ontology design best practices of reuse and alignment with standards an reference ontologies. Others, including the OntoSensor ontology [19], use the concepts defined in the OGC SensorML[8] standard as a basis. More recent proposals like [20] and [21], also consider the OGC Observations and Measurements (O&M) standard[9] to represent observations captured by sensor networks.

Recently, through the W3C SSN-XG group, the semantic web and sensor network communities have made an effort to provide a domain independent ontology, generic enough to adapt to different use-cases, and compatible with the OGC standards at the sensor and observation levels. The result, the SSN ontology [2], is based on the stimulus-sensor-observation design pattern [22] and the OGC standards.

[8] SensorML. `http://www.opengeospatial.org/standards/sensorml`
[9] OGC O&M: `http://www.opengeospatial.org/standards/om`

7 Conclusions and Future Work

We have described an approach for identifying the type of data from sensor data sources, using a symbolic representation of the time series slopes. We have shown how this representation can be used for enriching semantic sensor metadata. We have shown specific use cases of time series data classification, providing similarity measures, and metadata aggregation that can be queried in terms of high-level standard ontologies. Finally, we evaluated our approach with real-life datasets of the Swiss-Experiment project and AEMET.

We have shown through experimentation that this representation can be useful for balanced datasets, as the classification gets biased when there are small numbers of samples in the training set, for a particular type of data. Moreover, our results show that this representation can help grouping data of the same type, despite geographical locations, since it is based on the distribution of slopes of a linear approximation. Therefore, it can identify similarities of related types of data: e.g. *air temperature* and *soil temperature*. We have compared our characterization of sensor data with a competitive approach, and showed that for the chosen environmental datasets it effectively enables the extraction of semantic metadata.

The proposed approach, however, was evaluated within the same dataset, and in the future we will study its applicability in an inter-dataset classification. This framework could be used in the future for other tasks such as clustering, or for identifying simple patterns in streams of sensor data. Moreover, complex symbolizations consisting of sequences of slopes could be considered, which would represent more complete patterns that can be exploited. Also, we can consider building a more complex representation that includes not only the slopes information but also the value ranges, and even tags and labels provided the data publishers. This may enable a more complete and accurate extraction of metadata that enriches the growing Semantic Sensor Web. As a final future path, we may consider applying online execution of these techniques for real-time analysis.

References

1. Sheth, A., Henson, C., Sahoo, S.: Semantic sensor web. IEEE Internet Computing **12**(4) (2008) 78–83
2. Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C semantic sensor network incubator group. Journal of Web Semantics **(In press)** (2012)
3. Calbimonte, J.P., Jeung, H., Corcho, O., Aberer, K.: Semantic sensor data search in a large-scale federated sensor network. In: Proc. 4th International Workshop on Semantic Sensor Networks. (2011) 14–29
4. Buragohain, C., Shrivastava, N., Suri, S.: Space efficient streaming algorithms for the maximum error histogram. In: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, Ieee (2007) 1026–1035
5. Lin, J., Keogh, E.J., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**(2) (2007) 107–144

6. Xing, Z., Pei, J., Keogh, E.J.: A brief survey on sequence classification. SIGKDD Explorations **12**(1) (2010) 40–48
7. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**(3) (2001) 263–286
8. Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Transactions on Database Systems (TODS) **27**(2) (2002) 188–228
9. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. Data mining in time series databases **57** (2004)
10. Kasetty, S., Stafford, C., Walker, G., Wang, X., Keogh, E.: Real-time classification of streaming sensor data. In: Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on. Volume 1., IEEE (2008) 149–156
11. Möller-Levet, C., Klawonn, F., Cho, K., Wolkenhauer, O.: Fuzzy clustering of short time-series and unevenly distributed sampling points. Advances in Intelligent Data Analysis V (2003) 330–340
12. Zoumboulakis, M., Roussos, G.: Escalation: Complex event detection in wireless sensor networks. In: Proceedings of the 2nd European conference on Smart sensing and context, Springer-Verlag (2007) 270 – 285
13. Payam Barnaghi, Frieder Ganz, C.H., Sheth, A.: Computing perception from sensor data. In: Proceedings of the 2012 IEEE Sensors Conference (to appear). (2012)
14. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.J.: Querying and mining of time series data: experimental comparison of representations and distance measures. PVLDB **1**(2) (2008) 1542–1552
15. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning ICML 06. Volume 150., ACM Press (2006) 1033–1040
16. Geurts, P.: Pattern extraction for time series classification. Principles of Data Mining and Knowledge Discovery (2001) 115–127
17. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2009) 947–956
18. Compton, M., Henson, C., Lefort, L., Neuhaus, H., Sheth, A.: A survey of the semantic specification of sensors. In: Proc. 2nd International Workshop on Semantic Sensor Networks. (2009) 17
19. Russomanno, D., Kothari, C., Thomas, O.: Sensor ontologies: from shallow to deep models. In: Proc. 37th Southeastern Symposium on System Theory. (2005) 107–112
20. Barnaghi, P., Meissner, S., Presser, M., Moessner, K.: Sense and sensability: Semantic data modelling for sensor networks. In: Proceedings of the ICT Mobile Summit. (2009)
21. Compton, M., Neuhaus, H., Taylor, K., Tran, K.: Reasoning about sensors and compositions. In: SSN. (2009)
22. Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In: SSN. (2010) 7–11